可公开验证的属性基数据可恢复性证明方案 *

任 燕 [1,2]，唐春明 [2]

(1. 运城学院 数学与信息技术学院，山西 运城 044000; 2. 广州大学 数学与信息科学学院 广东省信息安全技术重点实验室，广州 510006)

摘 要：数据可恢复性证明方案可以有效解决用户将数据存储在不诚实的服务器上时，需要对数据的完整性进行验证的问题。针对目前存在的大部分方案都是使用的基于身份的密码体制的问题，采用更直观灵活的基于属性的密码体制设计了基于属性的数据可恢复性证明方案。给出了方案的相关定义、安全模型和具体的构造，同时证明了方案的正确性和安全性。

关键词：云计算；基于属性密码学；数据可恢复证明；公开验证

中图分类号：TP309.2 doi: 10.19734/j.issn.1001-3695.2018.06.0557

Attribute-based proof of retrievability with public verifiability

Ren Yan[1, 2], Tang Chunming[2]

(1. *School of Mathematics & Information Technology*, *Yuncheng University*, *Yuncheng Shanxi* 044000, *China*; 2. *Key Laboratory of Information Security*, *Guangzhou University*, *Guangzhou* 510006, *China*)

Abstract: When the user stores the data on an dishonest server, the integrity of the data needs to be verified, so the proof-of-retrievability (POR) system is proposed. Aiming at the problem of identity-based cryptosystem used in most of the existing solutions, a more intuitive and flexible attribute-based cryptosystem is adopted to design an attribute-based data recoverability proof scheme. This paper firstly proposed definition and model of an attribute-based proof of retrievability with public verifiability, then, constructed an attribute-based proof of retrievability with public verifiability, finally, proved its correctness and security.

Key words: cloud computing; attribute-based cryptography; proof-of-retrievability; public verifiability

0 引言

在数据外包存储服务中，如何使数据拥有者有效而且安全地验证存储服务器是否正确存储了他们的数据是非常重要的。为了解决这个问题，Juels 等人[1]首次正式地提出了数据可恢复性证明(proof of retrievability,POR)的概念和方案。在数据可恢复性证明的方案中，存储服务器必须向用户证明他们确实正确存储了用户的数据，并且让用户确信他可以恢复先前存储在服务器上的文件。文献[1]中所提的方案中，通过将一个伪装块隐藏在不变的文件块中来发现服务器对数据的修改，同时这个方案的通信代价是与每个文件块中元素数目成线性关系的。在此之后,很多学者对数据可恢复性证明的方案进行了研究。近年来的研究成果主要有：文献[2,3]中提出具有常数通信的 POR 方案；朱岩等人[4]在交互式证明系统的标准模型下提出了交互式可恢复性证明的形式化定义，并提出了一个实用的零知识可恢复性证明。该方案在 Diffie-Hellman 假设下被证明具有完备性、合理性和零知识性,该证明是通过建立多项式时间的知识提取器实现的，且该协议只需要发送固定大小的数据量就能够实现承诺、挑战、响应过程，并最大限度地减少网络通信。因此，该方案可用于大范围分布存储系统中实现大尺寸文件的公共远程验证。文献[5]提出了两个有效的动态的数据可恢复性证明方案，一个是私有验证，一个可以公开验证。文献[6~9]分别提出了几个动态可公开验证的数据可恢复性证明方案。

目前存在的大部分方案都是使用的基于身份的密码体制的思想,本文考虑将基于属性的密码体制用在数据可恢复性证明中，与基于身份的密码体制相比,基于属性的密码体制更直观。例如,从签名来说,某人使用一个基于身份的签名对消息进行签名后，验证者可以证实该消息的签名确实来自这个人，但是对于这个人所拥有的权限和社会职能却一无所知；而基于属性的签名中验证者可以检验签名是否为相应的属性的拥有者的签名,所以可以知道签名者的权限和职能，并且对于签名者的身份具有匿名性。如果用现实中的盖章来说明基于身份的签名与基于属性的签名的不同，则基于身份的签名就像是盖私章，而基于属性的签名就像是盖公章，私人章只能说明负责人是谁，而公章则可以表明颁发此签名的单位机构或者属性。在实际生活中,公章显然比签名章更具可信度。

本文首次提出了基于属性的可公开验证的数据可恢复性证明方案的的定义和安全模型,实现了基于属性的可公开验证的数据可恢复性证明方案的构造,给出了方案的正确性和安全性证明。在本文的方案中,用户只知道文件所有者的属性,而不知道关于文件所有者的任何的身份信息。

1 预备知识

1.1 双线性映射

设 G_1 , G_2 是两个循环乘法群, G_1 、 G_2 的阶均为素数 q 。设

收稿日期：2018-06-29；**修回日期：**2018-09-12 **基金项目：**广东省信息安全技术重点实验室开放课题基金资助项目（GDXXAQ2016-05）；山西省自然科学基金资助项目（20161D021014）；运城学院博士启动基金资助项目（2014-03）

作者简介：任燕（1982-）女，山西运城人，博士研究生，主要研究方向为密码学、信息安全（renyan-2000@163.com）；唐春明(1972-)，男，湖南怀化人，教授，博导，主要研究方向为云计算、密码学、信息安全等.

$e: G_1 \times G_1 \to G_2$ 为一个双线性映射.假定在 G_1 , G_2 上的离散对数问题(DLP 问题)都是困难的,则双线性映射满足以下性质：

a)双线性性。对任意的 P，$Q \in G_1$ 和所有的 $a,b \in Z_q$,有 $e(P^a, Q^b) = e(P,Q)^{ab}$ 。

b)非退化性。存在 $P,Q \in G_1$ 使得 $e(P,Q) \neq 1$ 。

c)可计算性。对于 $P,Q \in G_1$ 存在一个高效的算法计算 $e(P,Q)$ 。

1.2　拉格朗日插值定理

设 $f(x)$ 为 x 的一个次数为 n 的多项式 f 的函数,如果给定多项式 $n+1$ 个不同点 $(x_i, f(x_i))$,则通过式(1)能唯一确定任意一个 x 所对应的多项式 $f(x)$ 值:

$$f(x) = \sum_{i=1}^{n} f(x_i)(\prod_{1 \le k \ne i \le n} (x - x_k)/(x_j - x_k)) \tag{1}$$

对于式(1)中可以定义拉格朗日系数 $\Delta_{i,s}$,其中 $i \in Z_p$,集合 s 中的元素取自 Z_p ：

$$\Delta_{i,s}(x) = \prod_{i \in s, j \ne i} \frac{x-j}{i-j}$$

1.3　本方案依赖以下困难性问题

a)离散对数问题(DLP)。设 G 是一个阶为素数 p 的加法循环群,对 $P,Q \in G$,不存在多项式时间的算法可以以不可忽略的优势计算出 $n \in Z_q$,使得 $P = nQ$ 。

b)双线性对求逆问题。设 e 是一个双线性映射, G 是一个阶为素数 p 的加法循环群,对 $P \in G$ 和 $a = e(P,Q)$,不存在多项式时间的算法可以以不可忽略的优势计算出 $Q \in G$ 。

c)计算 Diffie-Hellman 问题(CBDH)。设 G 是一个阶为素数 q 的乘法循环群,对 $a,b \in Z_q$ 和 $(P, P^a, P^b) \in G_1$,不存在多项式时间的算法可以以不可忽略的优势计算出 $e(P,P)^{ab}$ 。

2　基于属性的可公开验证数据可恢复性证明方案的定义和安全模型

一般地，一个基于属性的可公开验证的数据可恢复性证明方案有三个参与实体：数据拥有者、客户和云服务提供商。数据拥有者对数据进行采集，为了提高数据的鲁棒性,数据拥有者把采集到的数据进行纠错码编码，然后将编码后的数据存储到云端。满足数据拥有者指定属性的客户可以访问编码后的数据并对数据的完整性进行检验.为了验证数据的完整性，客户端生成一个挑战信息并把它发送给云端，然后云端对客户端所选择的文件块进行计算证明响应。收到证明后,客户端可以通过验证算法验证数据的完整性。

2.1　基于属性的可公开验证的数据可恢复性证明的形式化定义

令 ω 是可能的属性集合, ω 上的一个断言实际上是一个输入为关于属性 ω 的布尔函数。当 $\Gamma(\omega')=1$ 时，本文称属性集合 $\omega' \subseteq \omega$ 满足一个断言 Γ 。方案的主要步骤如下所示：

算法

setup　生成主公钥和主密钥

keygen　生成属性集合密钥

outsource　1、编码

　　2、计算认证标签

proof　用户发起挑战时服务器为用户生成

verify　用户验证文件是否完整保存

综上，一个基于属性的可公开验证的数据可恢复性证明方案语义定义如下：

定义 1　形式上一个基于属性的可公开验证的数据可恢复性证明方案由初始化算法 Setup、密钥生成算法 Keygen、外包存储算法 Outsource、证明算法 Proof 和验证算法 Verify 组成：

a)Setup。该算法的输入为安全参数 k，输出为系统的主公钥 pk 和主私钥 mk 。

b)Keygen。属性中心为用户所选择的属性集合生成密钥。

c)Outsource。外包存储分为两个阶段。第一个阶段为编码：输入为用户的私钥和文件 M，输出一个大小为 n 的编码文件 $\tilde{M}$,第二个阶段为认证标签的计算：输入为编码文件 $\tilde{M}$，输出为标签。

d)Proof。输入为系统参数、断言 Γ 、满足 $\Gamma(\omega')=1$ 的含有 l 个元素的属性集合 $S \subseteq \omega$ 和编码后的文件消息 $\tilde{M}$，输出为证明响应 prf　。

e)Verify。输入为系统参数、证明响应 prf，该算法的输出为 accept 或者 reject。

2.2　安全模型

在这一节描述基于属性的可公开验证的数据可恢复性证明方案的安全模型。与文献[11~13] 一样，本文认为存储服务提供者是不可信且可能是恶意的，要求方案必须满足正确性和合理性。

定义 2　正确性。

若对任意的由如上定义的算法(KeyGen,Outsource, Proof) 生成的有效证明，验证算法都输出 accept，则称一个基于属性的可公开验证的数据可恢复性证明方案是正确的。

对于合理性，若任意恶意的云服务提供者可以生成一个证明，通过验证算法，即验证算法相信他确实正确的存储了文件 $\tilde{M}$，则他必须要有证明需要的 $\tilde{M}$ 。由 [10-12] 中提出的关于合理性的安全模型。类似地，本文给出下面的游戏。

a) Setup。挑战者运行 Setup 算法，获得公私钥对 (pk, sk)，并将 pk 发送给敌手。

b) Outsource。敌手选择一个文件 M，将它发送给挑战者。挑战者运行 Outsource 算法，用编码的文件来响应 $\tilde{M}$ 。

c) Proof。

(a) 挑战随机生成一个挑战信息，并将它发送给敌手；

(b) 因为敌手可能丢失或者修改文件 $\tilde{M}$ 的一部分，敌手首先随机生成一个文件 $\tilde{M}$;

(c) 敌手通过运行任意的算法生成一个证明 prf，并将 prf 发送给挑战者。

d) Verify。挑战者运行验证算法来检验 prf .当且仅当敌手可以对文件 $\tilde{M}$ 生成的证明 prf 且挑战者通过运行算法输出 accept 的时候，本文称敌手赢得游戏。

定义 3　合理性。

若对任意的概率多项式时间的敌手赢得上述游戏的概率是可忽略的，则称一个基于属性的可公开验证的数据可恢复性证明方案是合理的。

3　可公开验证的基于属性的可恢复性证明方案

假设集合 U 中有 l 个属性. U 中每个元素对应于 Z_p 中一个唯一整数。在这个方案中文件 M 的规模是任意长度的比特串。

现在描述基于属性的可公开验证的数据可恢复性证明方案，它支持所有的断言 Γ 。特别地，对门限值 d 有

$$\Gamma_{d,\omega^*}(\omega') = \begin{cases} 1, & |\omega' \cap \omega^*| > d \\ 0, & \text{其他} \end{cases}$$

具体的构造如下：

1) 初始化 (Setup)

Setup 阶段由以下几步完成：

a) 定义属性集合 ω ,简单起见令 $|\omega| = l$ 且本文可以取 Z_p

的前 l 个元素来做为这个集合,即： $1,2,\cdots,l(\mathrm{mod})p$ 。

b) 设 G_1 是阶为素数 p 的乘法循环群,随机选择一个生成元 $g\in G_1$,随机选取 $x,\in Z_p^*$， 并令 $X=g^x$ 。

c) 随机选择一个长度为 n 的向量 $U=(u_1,u_2,...,u_n)$,这里的 u_i 是群 G_1 中的元素且 $n<l$ 。

则公共参数为： $params=(g,X,d,\ U)$ 。

主密钥为 x 。

2) 密钥生成 (key generation)

按照如下步骤为用户设定的的属性集合 ω 生成私钥。

a) 选择一个 $d-1$ 阶的多项式 $q(x)$ 满足 $q(x)=0$ 。

b) 对每个 $i\in\omega$ 计算： $d_i=g^{q(i)}$ 。

c) 对每个 $i\in\omega$ 输出 d_i 做为私钥。

3) 外包存储

a) 给定一个数据文件 M ，利用纠错码得到编码后的文件 $\tilde{M}$ 。

b) 将编码后的文件 $\tilde{M}$ 分成 n 个块, $\tilde{M}=(\mu_1,\mu_2,...,\mu_n)$ 其中 $\mu_i\in\{0,1\}^t$ 。

c) 对每个数据块按照如下的步骤完成认证标签计算：

(a) 从 Z_p^* 中随机选择一个文件名 name；

(b) 随机选取的一个值 $s\in Z_p$， 并公开 g^s ；

(c) 计算： $\sigma_i=d_i^{\Delta_{i,S}(0)}(u^{H(name\|i)}\prod_{i=1}^{n}u_i^{\mu_i})^s$,这里 $i\in\{1,2,\cdots,\ n\}$, S 是属性集合 ω 的一个子集,且 $|S\cap\omega|\ge d$, u 为客户端随机选择的一个数，然后将 $\tilde{M}$ 和 σ_i 外包给云服务器存储。

4) 验证挑战 (verify challenge)

客户端为了验证服务器是否正确存储了文件,随机选择属性集合 U 的 d 个元素的子集 S' 和 $r\in Z_p^*$ 并将它们作为挑战发送给服务器。

5) 证明生成 (ProofGen)

服务器收到验证挑战后计算: $\sigma=\prod_{i=1}^{n}\sigma_i^r$, $\psi=\prod_{i=1}^{n}u_i^{\mu_i}$.则生成的证明 $prf=(\sigma,\psi)$.然后服务器将证明发送给客户端。

6) 验证 (verify)

收到证明 prf 后，客户端首先计算: $\eta_i=u^{H(name\|i)}$, $\eta=\prod\eta_i$ ，然后验证如下等式是否成立来验证服务器是否正确存储了文件,且未对文件进行窜改：

$$e(\sigma,g)=e(\eta^r,g^s)e(\psi^r,g^s)e(X,g^r)$$

4　正确性和安全性分析

4.1　正确性分析

定理 1 方案的验证过程是正确的。

证明

$$e(\sigma,g)=e(\prod_{i=1}^{n}\sigma_i^r,g)=$$

$$e(\prod_{i=1}^{n}d_i^{\Delta_{i,S}(0)r}(u^{H(name\|i)}\prod_{i=1}^{n}u_i^{\mu_i})^{rs},g)=$$

$$e(\prod_{i=1}^{n}d_i^{\Delta_{i,S}(0)r},g)e((\prod u^{H(name\|i)})^{rs},g)\bullet e((\prod_{i=1}^{n}u_i^{\mu_i})^s,g^r)=$$

$$e(\eta^r,g^s)e(\psi^r,g^s)e(X,g^r)$$

4.2　安全性分析

定理 2　本文方案中的证明 prf 在 CBDH 假设下满足合理性,即它是不可伪造的。

证明　假设一个概率多项式时间的敌手 F 可以以不可忽略的优势赢得合理性游戏,则本文可以构建一个算法 A 来解决 CDH 问题。即算法 A 在给定 g， g^r , g^x ,时,可计算出 $e(g,g)^{rx}$.具体过程如下:

1)初始化 (Setup)

F 输出挑战断言,即 l 个元素的属性集合 ω^* 的门限为 d 的函数.　令 $g_1=g^x$ 。

2)密钥生成 (key generation)

A 可以对私钥进行查询.按照如下方式模拟生成属性 ω 的私钥:

a)随机选取 $s_1\in Z_p^*$,令 $s=-y+s_1$ 。

b)定义三个集合 $\Gamma=\omega^*\cap\omega$, Γ' ， $S=\dot{\Gamma}\cup\{0\}$ ，这里 Γ' 满足 $\Gamma\subseteq\Gamma'\subseteq\omega$ 。

若 $i\in\Gamma'$ 则 $d_i=g_1^{\tau_i}$,这里的 τ_i 是在 Z_p 中随机选取。

若 $i\notin\Gamma'$,令,则 $d_i=g_1^{\sum_{(j\in\Gamma'}\Delta_{j,S}(i)q(j)+\Delta_{0,S}(i)q(0)}$

由 $q(i)=\sum_{(j\in\Gamma'}\Delta_{j,S}(i)q(j)+\Delta_{0,S}(i)q(0)$

可知 A 正确模拟了私钥。

因此有 $g_1^{q(i)}=g_1^{\sum_{(j\in\Gamma'}\Delta_{j,S}(i)q(j)+\Delta_{0,S}(i)q(0)}$ 。

3)外包存储模拟

A 可以用属性集合 ω 完成对消息 m 的外包存储。

若 $|\omega\cap\omega^*|\ge d$,则 A 通过如下过程模拟签名：

选择随机的 $s\in Z_q$,按照正常的方式得到属性集合 ω 对消息 m 的外包存储。

若 $d\ge|\omega\cap\omega^*|\ge k$,则 A 可以如下模拟签名：

$$\sigma_i'=g_1^x(u^{H(name\|i)}\prod_{i=1}^{n}u_i^{\mu_i})^s\ 。$$

4)验证挑战模拟(verify challenge)

客户端随机选择属性集合 U 的 d 个元素的子集 S'' 和 $r'\in Z_p^*$ 并将它们发送给服务器。

5)证明生成模拟 (ProofGen)

服务器计算： $\sigma'=\prod_{i=1}^{n}\sigma_i^{r''}$ ， $\psi'=\prod_{i=1}^{n}u_i^{\mu_i}$ ，则生成的证明 $prf'=(\sigma',\psi')$.然后将证明发送给客户端。

验证过程模拟;

收到证明 prf' 后，客户端首先计算: $\eta_i=u^{H(name\|i)}$, $\eta=\prod\eta_i$ ，然后如下等式成立:

$$e(\sigma',g)=e(\eta^r,g^s)e(\psi'^r,g^s)e(X,g^r)$$

这里 $prf'=(\sigma',\psi')\ne prf=(\sigma,\psi)$

由此可以得到:

$$e(\sigma,g)=e(\eta^r,g^s)e(\psi^r,g^s)e(X,g^r)\qquad(1)$$

$$e(\sigma',g)=e(\eta^r,g^s)e(\psi'^r,g^s)e(X,g^r)\qquad(2)$$

用式（1）除以式（2）得到 $\frac{e(\sigma,g)}{e(\sigma',g)}=\frac{e(\psi^r,g^s)}{e(\psi'^r,g^s)}$

下面分析第一种情形:

若 $\sigma\ne\sigma'$, $\psi=\psi'$ ，

则： $e(\sigma,g)=e(\sigma',g)$

即

$$e(\sigma',g)=e(\sigma,g)=e(\prod\sigma_i^r,g)=$$

$$e(\prod\ d_i^{\Delta_{i,S}(0)r}(u^{H(name\|i)}\prod u_j^{\mu_j})^{sr},g)=$$

$$e(\prod_{i=1}^{n}d_i^{\Delta_{i,S}(0)r},g)e((\prod u^{H(name\|i)})^{rs},g)\bullet e((\prod_{i=1}^{n}u_i^{\mu_i})^s,g^r)=$$

$$e(\eta^r,g^s)e(\psi^r,g^s)e(X,g^r)$$

由于 $\psi=\psi'$，且 $\eta_i=u^{H(name\|i)}$, $\eta=\prod\eta_i$

所以敌手可以计算 $e(g^x,g^r)=\frac{e(\sigma',g)}{e(\eta^r,g^s)e(\psi^r,g^s)}$

而 x,r 对于敌手是保密的，所以敌手可以解决 CBDH 问题。

下面分析第二种情形：

若 $\sigma=\sigma'$, $\psi\neq\psi'$

则
$$e(\psi',g)=e(\psi,g)$$

即
$$e(\psi'^r,g^s)=e(\psi^r,g^s)=e((\prod u_j^{\mu_j})^r,g^s)$$

所以，敌手可以计算 $e((\prod u_j^{\mu_j})^r,g^s)=e(\psi'^r,g^s)$

而 s,r 对于敌手是保密的，所以敌手可以解决 CBDH 问题。

下面分析第三种情形：

若 $\sigma\neq\sigma'$, $\psi\neq\psi'$

则
$$\frac{e(\sigma,g)}{e(\sigma',g)}=\frac{e(\psi^r,g^s)}{e(\psi'^r,g^s)}$$

即
$$e(\sigma',g)=\frac{e(\psi'^r,g^s)e(\sigma,g)}{e(\psi^r,g^s)}=$$
$$\frac{e(\psi'^r,g^s)e(\eta^r,g^s)e(\psi^r,g^s)e(X,g^r)}{e(\psi^r,g^s)}=$$
$$e(\eta^r,g^s)e(\psi'^r,g^s)e(X,g_1^{\ r})$$

由于 $\eta_i=u^{H(name\|i)}$, $\eta=\prod\eta_i$

所以，敌手可以计算： $e(g^x,g_1^{\ r})=\frac{e(\sigma',g)}{e(\eta^r,g^s)e(\psi'^r,g^s)}$

而 x,r 对于敌手是保密的，所以敌手可以解决 CBDH 问题。

5 结束语

本文首次将基于属性的签名思想用在数据可恢复性证明中，提出了基于属性的可公开验证的数据可恢复性证明方案的的定义和安全模型,实现了基于属性的可公开验证的数据可恢复性证明方案的构造,给出了方案的正确性和安全性证明。

参考文献：

[1] Juels A, Jr Kaliski B S. PORs: proofs of retrievability for large files [C]//Proc of the 14th ACM Conference on Computer and Communications Security. New York:ACM Press, 2007: 584-597.

[2] Xu, Jia, Chang E C. Towards efficient provable data possession [J]. IACR Cryptology ePrint Archive, 2011, 2011: 574.

[3] Yuan, Jiawei, Yu Shucheng. Proofs of retrievability with public verifiability and constant communication cost in cloud [C]// Proc of International Workshop on Security in Cloud Computing. 2013: 19-26.

[4] Zhu, Yan, Wang Huaixi, Hu Zexing, *et al.* Zero-knowledge proofs of retrievability [J]. Science China Information Sciences, 2011, 54 (8): 1608-1617.

[5] Shi E, Stefanov E, Papamanthou C.Practical dynamic proofs of retrievability[C]//Proc of ACM Conference on Computer and Communications Security.New York:ACM Press, 2013: 325–336.

[6] Beng T C, Hijazi M H A, Lim Y , *et al.* A survey on Proof of Retrievability for cloud data integrity and availability: cloud storage state-of-the-art, issues, solutions and future trends[J]. Journal of Network and Computer Applications,2018,110(2018):75-86.

[7] Ren Zhengwei, Wang Lina, Wang Qian,*et al*. Dynamic proofs of retrievability for coded cloud storage systems[J]. IEEE Trans on Services Computing, 2018, 11(4), 685-698.

[8] Fu Anmin, Li Yuhan, Yu Shui, *et al*. DIPOR: an IDA-based dynamic proof of retrievability scheme for cloud storage systems [J]. Journal of Network and Computer Applications, 2018,104: 97-106.

[9] Sengupta B, Ruj S. Efficient proofs of retrievability with public verifiability for dynamic cloud storage [J]. IEEE Trans on Cloud Computing, 2017.

[10] Cash D, Küpçü A, Wichs D. Dynamic proofs of retrievability via oblivious RAM [J]. Journal of Cryptology, 2017, 30 (1): 22-57.

[11] Juels A, Kaliski Jr B S. PORs: Proofs of retrievability for large files [C]//Proc of the 14th ACM Conference on Computer and Communications Security. ACM, 2007: 584-597.

[12] Shacham H, Waters B. Compact proofs of retrievability[C]//Advances in Cryptology-Asiacrypt. Berlin:Springer, 2008: 90-107.